Dagstuhl Report No. 232

Deduction

Ulrich Furbach
Universität Koblenz-Landau
Fachbereich Informatik
Rheinau 1
56075 Koblenz
Germany
E-mail: uli@uni-koblenz.de

Harald Ganzinger
Max-Planck-Institut für Informatik
Programming Logics Group
Im Stadtwald
D-66123 Saarbrücken
Germany
E-mail: hg@mpi-sb.mpg.de

Ryuzo Hasegawa
Department of Electronics
Kyushu University 36,
Fukuoka 812,
Japan
E-mail: hasegawa@ele.kyuhsu-u.ac.jp

Deepak Kapur
University of New Mexico
Dept. of Computer Science
Farris Eng. Center # 339
NM87131 Albuquerque, USA
E-mail: kapur@cs.unm.edu

This report[1] covers the seminar no. 99091 on *Deduction*, held at Dagstuhl, Germany during February 28–March 5, 1999. This seminar was organized by U. Furbach (Koblenz, Germany), H. Ganzinger (Saarbrücken, Germany) and D. Kapur (Albuquerque, USA). It brought together about 50 researchers from various countries.

Dagstuhl, a place being developed exclusively for research activities in Computer Science, provides an excellent atmosphere for researchers to meet and exchange ideas. During this seminar we had 40 talks and a discussion on the focus programme 'Deduction'.

[1]Compiled by Jan Murray, Universität Koblenz-Landau.

Contents

1 A Note from the Organizers

Logic has become a prominent formal language and formalism for all of computer science. It serves in many applications such as in problem specification, program development, program transformation, program verification, hardware design and verification, consistency checking of databases, theorem proving, expert systems, logic programming, and so on and so forth. Its strength derives from the universality of the language as well as from the fundamental logical operations and relations. Logical manipulations as needed in all these applications are realized by mechanisms developed in the field of deduction which has produced a variety of techniques of great importance in all these applications.

All these research issues have been subject of a "Schwerpunktprogramm Deduktion" funded by the Deutsche Forschungsgemeinschaft.

During the last years successful research in this program has led to the development of high performance deduction systems, and to laying a broad basis for various applications.

This success of deduction can be observed within the international AI and computer science scene as well. Deduction systems recently have achieved considerable successes and even public press: it was a first-order theorem prover which first proved the Robbins algebra conjecture and even reached the New York Times Science section. But not only in proofing mathematical theorems, also in various other disciplines of AI, automated deduction made substantial progress. In planning, for example, it turned out that propositional theorem provers are able to outperform special purpose planning systems. This is remarkable, since it was considered folklore that planning requires specialized algorithms, which was only recently disproved by the development of propositional satisfiability testing methods which can now handle much larger planning problem sizes. A very similar development can be observed in the field of model based diagnosis.

It is the idea of this Dagstuhl-Seminar to bring together leading scientists in the area of Deduction from all over the world. By all participants the previous seminars in 1993, 1995 and 1997 were unanimously considered great successes. The Dagstuhl-Seminar Reports No. 58, 110 and 170 of these seminars, together with this one, reflect the development of the entire discipline in the 90th.[2]

Ulrich Furbach, Harald Ganzinger, Ryuzo Hasegawa, Deepak Kapur

[2]They are still available in the Dagstuhl Office or on the Dagstuhl Web-pages.

2 Abstracts of the Talks

Jürgen Avenhaus, Universität Kaiserslautern, Germany (joint work with C. Loria-Saenz, A. Middeldorp, and T. Yamada)

Logicality of Conditional Rewrite Systems

Conditional rewriting provides a useful framework for the study of a wide range of problems in computation and programming. Here we investigate the logical strength of conditional rewrite systems (CTRS). A CTRS is called logical if is has the same logical strength as the underlying conditional equational system. Logicality is important because it says that an equation is valid in all models of the underlying equational system iff it is provable by rewriting. Different types of conditional rewrite systems are studied in the literature. They differ in the way the conditions are evaluated for performing a rewrite step. In a semi-equational CTRS the conditions are checked by allowing rewriting in both directions. In a join CTRS the conditions are checked by joinability. In an oriented CTRS the conditions are evaluated by rewriting from left to right. This allows one to cope with extra-variables and has some similarities with logic programming: Information is gathered while evaluating the conditions from left to right and the extra-variables in the right-hand side of a rule are bound this way. In the talk we study mainly two questions:

1. How to ensure logicality of an oriented CTRS?
2. How do the different rewrite relations compare for the same underlying equational system?

Franz Baader and Cesare Tinelli, RWTH Aachen, Germany, University of Illinois at Urbana-Champaign, USA

Deciding the Word Problem in the Union of Equational Theories Sharing Constructors

The main contribution of this work is a new method for combining decision procedures for the word problem in equational theories sharing "constructors." The notion of constructor adopted here has a nice algebraic definition and is more general than a related notion introduced in previous work on the combination problem.

Peter Baumgartner, Norbert Eisinger and Ulrich Furbach,
Universität Koblenz, Universit"at M"unchen, Germany

A Confluent Connection Calculus

This work is concerned with basic issues of the design of calculi and proof procedures for first-order connection methods and tableaux calculi. Proof procedures for these type of calculi developed so far suffer from not exploiting proof confluence, and very often unnecessarily rely on a heavily backtrack oriented control regime.

As a new result, we present a variant of a connection calculus and prove its *strong* completeness. This enables the design of backtrack-free control regimes. To demonstrate that the underlying fairness condition is reasonably implementable we define an effective search strategy. We show that with the new approach the search space can be exponentially smaller than those of current, backtracking-oriented proof procedures based on *weak* completeness results.

Bernhard Beckert, Universität Karlsruhe, Germany

Depth-first Proof Search without Backtracking for Free Variable Semantic Tableaux

In this talk, I analyse the problem of constructing a deterministic proof procedure for free variable tableaux that performs depth-first proof search without backtracking. As an example, I present a solution for a proof confluent version of clause tableaux.

The deterministic search strategy is based on a new notion of *regularity* to make sure that there are no "cycles" in the search (it is not possible to deduce the same formulae or sub-tableau again and again), and *weight orderings*, i.e., each tableau formula is assigned a "weight" in such a way that there are only finitely many different formulae (up to variable renaming) of a certain weight; thus, if tableau formulae with lesser weight are deduced first, then sooner or later each conclusion is added to all branches containing its premiss, i.e., the strategy is *fair*. In addition, to handle the destructiveness of free variable calculi, the strategy employs *reconstruction steps*. Immediately after a tableau expansion that destroys formulae, the expansion steps that are needed to recreate the destroyed formulae are executed.

The advantage of depth first proof search is that the information represented by the constructed tableaux increases at each proof step; no information is lost since there is no backtracking. In addition, considering similar tableaux or sequences of tableaux in different paths of the search tree is avoided.

The method is compatible with search space restrictions that preserve proof confluence of the calculus. Important examples for such restrictions include selection functions and the weak connectedness condition.

François Bry, Ludwig-Maximilians-Universität München

A proof method for minimal finite entailment

Databases can be formalized as finite models of first-order sentences representing static or dynamic integrity constraints, updates, and deduction rules. To this aim, a couple of simple and quite natural restrictions are necessary. Thus, the models to be considered are "term models", i.e. sort of Herbrand models with finite universes. The considered sentences must have "restricted quantifications" so as to formalize the Select-from-where structure of database queries and integrity constraint made necessary by the closed world assumption.

In this framework, the well investigated issue of query answering refers to the satisfaction of formulas in a given finite model. Other issues like constraint design, constraint redundancy, view update, constraint repair, and update soundenessm however, refer to finite satisfiability or entailment, or to relation minimal, finite satisfiability or finite entailment.

Since neither model minimality nor model finiteness can be axiomatized in first-order logic, standard proof methods cannot be applied for solving the afore-mentioned issues. In the presentation, a tableau method combining an extended rule for the elimination of existential quantifiers and a relation minimality test is presented. Finally, perspectives for further investigations are given.

Horatiu Cirstea and Claude Kirchner, LORIA - INRIA, France

The rewriting calculus

The rewriting calculus, also called ρ-calculus, is a new calculus that integrates in a uniform and simple settings first-order rewriting, λ-calculus and non-deterministic computations as well as their combination.

We describe the calculus from its syntax to its basic properties in the untyped case. We show how it embeds first-order rewriting and λ-calculus.

Finally we show how ρ-calculus can be used to give an operational semantics to the rewrite based language ELAN available at: www.loria.fr/ELAN.

Ingo Dahn, Universität Koblenz, Germany (joint work with A. Trybulec und Ch. Wernhard)

From Mizar to TPTP

Problems from applied deduction are almost always formulated in a typed language. Automated theorem provers, however, require untyped first order formulas as input.

The Mizar library contains formalized knowledge from a large variety of mathematical topics. It uses a sophisticated typed language. This language has proved to be a very comfortable tool for the formalization of mathematical knowledge.

We describe the translation of all theorems of the Mizar article "Rellations on Sets" written by E. Woronowicz into a set of untyped first order proof problems. These first order problems have been included into the TPTP library which is used as a standard test suite for automated theorem provers.

We concentrate on the encoding of specific features of the Mizar typre system into first order logic. These specifics include

- subtyping,
- the use of object parameters in type constructors,
- possibilities of casting the type of a term.

Experiments on the behaviour of some automated theorem provers on the resulting first order problem set are reported. In the last part of the talk we discuss some requirements to adapt automated theorem provers to the solution of typed proof problems.

Anatoli Degtyarev, Kiev Institute of Cybernetics and Uppsala University

A Specialized Form of Derivations in Calculi with both Equality and non-Equality Predicates.

We transfer the notion of regular derivations from sequent calculi with equality to superposition-based clause calculi. The regular form of derivations was introduced by Kanger, later it was used by Maslov to generalize the inverse method to predicate calculi with equality. According to this specialization, all applications of equality rules should be moved on the top of the proof so that they would precede all other steps in the proof (would follow all other steps in the sequent calculus proof). In clause calculi the regular specialization can be represented in terms of hierarchic first-order theories as introduced

by Bachmair, Ganzinger and Waldmann if we consider equality literals as non-base literals and non-equality literals as base literals. However, we cannot apply their model building proof technique to prove our completeness result because of ordering incompatibilities. Therefore the completeness of the specialization is proven by proof transformation. An essential point is the use of the basic strategy and a special form of basic factoring. The regular specialization of derivations in clause calculi has been used as a basis for the equality elimination method developed by Degtyarev and Voronkov for equality handling in the semantic tableau, connection and inverse methods.

Hans de Nivelle, Vrije Universiteit Amsterdam, Netherlands
(joint work with Harald Ganzinger)

A superposition based decision procedure for the guarded fragment with equality.

The guarded fragment is the subset of function free first order logic, in which all quantified subformulae are relativized by an atom that contains all free variables of the quantified subformula. The guarded fragment is decidable in DEXPTIME, also when equality is added. We show that the superposition calculus is a decision procedure for the guarded fragment with equality, when a suitable transformation to clausal normal form and a suitable ordering with selection function are used. The decision procedure is theoretically optimal, since it terminates in at most double exponential time. This talk generalizes earlier results by the first author, where a resolution decision procedure for the guarded fragment without equality was given.

Jörg Denzinger and Dirk Fuchs, Universität Kaiserslautern, Germany

Cooperation of Heterogeneous Provers

We present a methodology for achieving cooperation between already existing theorem provers employing different proof paradigms and/or different search controls, and using different but related logics. Cooperation between the provers is achieved by periodically interchanging clauses which are selected by so-called referees. By employing referees both on the side of a sending prover and a receiving prover the communication is both success- and demand-driven, which results in a rather small communication overhead and synergetical effects.

We report on experiments regarding the cooperation of the provers SPASS, SETHEO and DISCOUNT in domains of the TPTP library and with problems stemming from an application in software component retrieval. The

experiments show significant improvements in the number of problems solved as well as in the solution times. The methodology has also been successfully used in the area of optimization problems.

Uwe Egly, Technische Universität Wien, Austria (Joint work with S. Schmitt.)

On Intuitionistic Proof Transformations, their Complexity, and Application Constructive Program Synthesis

We present a translation of intuitionistic sequent proofs from a multi-succedent calculus LMC into a single-succedent calculus lj. The former gives a basis for automated proof search whereas the latter is better suited for proof presentation and program construction from proofs in a system for constructive program synthesis. Well-known translations from the literature have a severe drawback; they use cuts in order to establish the transformation with the undesired consequence that the resulting program term is not intuitive. We establish a transformation based on permutation of inferences and discuss the relevant properties with respect to proof complexity and program terms.

As an important result we show that LJ cannot polynomially simulate LMC (both without the *cut* rule), even in the propositional fragment.

H. Ganzinger, Chr. Meyer, and M. Veanes, MPI für Informatik, Saarbrücken, Germany

The Two-Variable Guarded Fragment with Transitive Relations

We consider the restriction of the guarded fragment without equality to the two-variable case where, in addition, binary relations may be specified as transitive. We show that (i) this very restricted form of the guarded fragment without equality is undecidable and that (ii) when allowing transitive relations to occur only in guards, the logic becomes decidable. The latter subclass of the guarded fragment is the one that occurs naturally when translating multi-modal logics of the type K4, S4 or S5 into first-order logic. We also show that the loosely guarded fragment with a single transitive relation is undecidable.

Pascal Gribomont, Université de Liège, Belgium

Towards Fully Automated Verification of Concurrent Systems

The problem of invariant validation for the control part of many concurrent and distributed programs reduces to the validity problem for classical propositional or first-order logic. The verification conditions tend to be very long, so specific simplification rules have to be applied before validation. This strategy works well in the propositional case. We propose an OBDD-based technique to extend this to the quantifier-free first-order case. The idea is to split the verification conditions into long but purely propositional formulas and short first-order quantifier-free formulas. However, many concurrent systems and their invariants involve a limited amount of quantifications; this typically occurs when the system contains an arbitrary number of symmetric processes. We propose a systematic technique for the elimination of such quantifications. This leads to a rather general verification technique, which applies to many "control-intensive" concurrent systems. As an illustration, we consider Ricart-Agrawala's algorithm, a distributed scheme for mutual exclusion in a computer network.

Reiner Hähnle, Universität Karlsruhe, Germany

Semantic Semantic Tableaux

Hyper tableaux and tableaux with selection function are combined to form a family of very general, saturation-based tableau calculi. Besides its generality, one benefit is a simple and uniform framework for virtually all known proof confluent ground tableau calculi. Another benefit is the first completeness argument obtained for certain extensions of model generation theorem proving.

Mitch Harris, University of Illinois, USA

Symmetry in Finite Model Checking

Concurrent programs can be annotated with temporal logic. Temporal logic can (sometimes!) be transformed into propositional calculus. Propositional calculus statements can be verified, but is coNP-complete. One heuristic that will speed up many instances is to verify only those instances that are essentially unique, with respect to permutation of the variables. We discuss and analyze this technique.

Ullrich Hustadt, Manchester Metropolitan University, Great Britain

Recent Results on Resolution-Based Decision Procedures for Modal Logics

This talk presents various resolution-based decision procedures for fragments of first-order logic corresponding to expressive multi-modal logic. All the results are based on the resolution framework of Bachmair and Ganzinger (1991). First, an ordering refinement of resolution is presented which solves the satisfiability problem for the class of DL-Clauses. This class of clauses corresponds to the multi-modal logic $\mathsf{K}_{(m)}(\mathsf{U}, \neg, \wedge, \vee, \smile)$ and its extension by the axiom schemata D, T, B, and W using an embedding of modal formulae into first-order logic by the relational translation. Second, we show that for $\mathsf{K}_{(m)}(\wedge, \vee, \smile)$ also a refinement of resolution based on a selection function provides a decision procedure. Furthermore, this decision procedure is able to p-simulate standard tableaux-based procedure for this modal logic. If we exclude factoring an redundancy elimination techniques, then the simulation results also holds in the opposite direction. In the third and final part, we turn to the semi-functional translation of modal logics. In particular, we present a refinement based on an ordering and a selection function that is able to solve the satisfiability problem for the modal logic K4 and its extensions by D and T. Together with the more straightforward decidability result for extensions of K by the axiom schemata B, D, T, and 5, our results now cover all the standard normal modal logics.

Deepak Kapur and G. Sivakumar, University of New Mexico, USA

Proving Associative-Communicative Termination Using Recursive Path-Ordering (RPO) Compatible Orderings

Developing path orderings for associative-commutative (AC) rewrite systems has been quite a challenge for some time. Compatibility with RPO is desirable, and this property helps in ordering the commonly encountered distributivity axiom as desired. For applications in theorem proving and constraint solving, a total ordership on (AC-equivalent) ground terms is often required. A generalization of our proposal presented at RTA'97 is discussed using which ordering schemas can be designed with the above desired properties that also handle general (non-ground) terms. The proposed definition allows flexibility (using different abstraction functions) in the way the candidates of a term are composed, thus leading to a family of distinct orderings on terms (from the same precedence relation on function symbols).

Claude Kirchner, Gilles Dowek, and Thérèse Hardin, LORIA - INRIA, France

Theorem Proving Modulo

"Theorem proving modulo" is a way to remove computational arguments from proofs by reasoning modulo a congruence on propositions. Such a technique, issued from automated theorem proving, is of wider interest because it permits to separate computations and deductions in a clean way. The first contribution of this work is to define a "sequent calculus modulo" that gives a proof theoretic account of the combination of computations and deductions.

The congruence on propositions is handled via rewrite rules and equations. In many cases, rewrite rules and equations just relate terms. Our second contribution is to show that applying rewrite rules directly to propositions is of prime interest in many examples.

The last contribution is to give a complete proof search method, called "Extended Narrowing and Resolution" (ENAR), for theorem proving modulo such congruences. The completeness of this method is proved with respect to the provability in sequent calculus modulo.

An important application is that higher-order logic can be presented as a theory modulo. Applying the Extended Narrowing and Resolution method to this presentation of higher-order logic subsumes full higher-order resolution.

Wolfgang Küchlin and Alfons Geser, Universität Tübingen, Germany

Structured Formal Verification of a Fragment of the IBM S/390 Clock Chip

With the increasing complexity of hardware, it becomes increasingly important to eliminate logical defects early. By the fact that hardware is organized along Boolean algebra, propositional reasoning is a mandatory part of Formal Verification. The now classical approach is *symbolic model checking*. Model checking determines the set of states of a given finite automaton which satisfy a given temporal or modal formula. The basic shortcoming of model checking, in our view, is its rigid language and the absence of structure.

Therefore, we pursue another approach: a combination of *term rewriting* and *propositional reasoning*. We switched from term rewriting to the related term graph rewriting for its ability to adequately represent hardware structure. The particular progress presented in this paper is a theorem that justifies a complete separation of the propositional reasoning part from the term graph rewriting part.

In our approach the verification problem breaks down into three steps:

1. Compilation of the netlist into modules of a typed term graph rewriting system. Sequential behaviour is modelled by deterministic Mealy automata.

2. The formal requirements are rewritten into a normal form, which is a term graph representing a propositional formula.

3. The propositional formula is evaluated by any appropriate method. In our experiments we used ordered functional decision diagrams. If the result represents 1 then the answer is "yes", else a counterexample may be constructed mechanically on request.

We illustrate these steps at a case study: a fragment of the IBM S/390 Clock Chip.

Alexander Leitsch, TU Vienna, Austria(joint work with Matthias Baaz)

Proof Transformation by Resolution

A new cut-elimination method for Gentzen's LK is defined. First cut-elimination is generalized to the problem of redundancy-elimination. Then the elimination of redundancy in LK-proofs is performed by a resolution method in the following way: A set of clauses $\mathcal{C}$ is assigned to an LK-proof ψ and it is shown that $\mathcal{C}$ is unsatisfiable. A resolution refutation of $\mathcal{C}$ then serves as a skeleton for an LK-proof ψ' with atomic cuts; ψ' can be constructed from the resolution proof and ψ by a projection method. Finally the atomic cuts are eliminated and a cut-free nonredundant proof is obtained. The complexity of the method is analyzed and it is shown that a nonelementary speed-up over Gentzen's method can be achieved. As the main complexity lies in the search for a resolution proof, the new method can take advantage of the strong search methods and redundancy deletion in automated deduction.

Reinhold Letz, TU München, Germany

Incompatibilities of Tableaux and Matings Refinements

We discuss central refinements of the tableaux and the matings framework in automated deduction and address the question whether a combination is possible. The general motivation for this is that both paradigms emphasize different aspects of proof theory. While a tableau represents the natural model of a dynamically-oriented path enumeration procedure, matings take

a more static and global look on the respective formulae. As has already been demonstrated, one can improve proof search when combining both methods.

In tableaux, the connectedness and the regularity condition have proven as indispensible for redundancy elimination. In the matings framework, on the other hand, the notion of the minimality of a mating is essential. The main result presented in this talk is that the mentioned three refinements are not compatible. More precisely, for certain unsatisfiable formulae, there exist no regular closed connection tableaux whose matings are minimal. This result prevents the general use of the minimality of matings in tableaux.

Christopher Lynch, Clarkson University, Potsdam, USA (joint work with Christelle Scharff)

Constrained Completion modulo E with Simplification

We give a new inference system for Basic Completion modulo an equational theory E with simplification that we call *BCIC-E*. Our system has all the advantages of Basic E-completion, in that E-unification problems are stored as constraints instead of being applied. But it has several advantages over previous work on Basic E-completion. One advantage is that we allow simplification where previous inference systems did not. Another advantage is that we only need to check the satisfiability of equational constraints instead of solving them. A third feature of our method is that simplification is presented concretely. For completeness we need to allow inferences into constraints, but we do this without losing the most important benefits that come from Basic E-completion.

Fabio Massacci, Università di Roma "La Sapienza", Italy

Automated Reasoning and the cryptanalysis of the US Data Encryption Standard

Providing formal assurance is a key issue for cryptographic algorithms. Yet, automated reasoning tools have only been used for the verification of security protocols, and almost never for the verification and cryptanalysis of the cryptographic algorithms on which those protocols rely.

In this talk I will claim that one can use logic for encoding the low-level properties of state-of-the-art cryptographic algorithms and then use automated theorem proving for reasoning about them. Let us call this approach *logical cryptanalysis*. In this framework, finding a model for a formula encoding an algorithm is equivalent to finding a key with a cryptanalytic attack. Other important properties can also be captured.

Here I present a case study on the U.S. Data Encryption Standard (DES) and discuss how to obtain a manageable encoding of its properties and how a number of SAT provers performs on the encoding of the DES.

Erica Melis and Jörg Siekmann, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken, Germany

Knowledge-Based Proof Planning

Knowledge-based proof planning is a new paradigm in automated theorem proving (ATP) which swings the motivational pendulum back to its AI-origins in that it employs and improves upon many AI-principles and techniques such as hierarchical planning, knowledge representation in frames and control-rules, constraint solving, tactical and meta-level reasoning. It differs from traditional search-based techniques in ATP not least with respect to its level of abstraction: the proof of a theorem is planned at an abstract level and an outline of the proof is found. This outline, i.e. the abstract proof plan, can be recursively expanded and it will thus construct a proof within a logical calculus. The plan operators represent mathematical techniques familiar to a working mathematician. While the knowledge of a domain is specific to the mathematical field, the representational techniques and reasoning procedures are general-purpose. The general-purpose planner makes use of this mathematical domain knowledge and of the guidance provided by declaratively represented *control-rules* which correspond to mathematical intuition about how to prove a theorem in a particular situation. These rules provide a basis for *meta-level reasoning* and goal-directed behaviour. We demonstrate our approach for the mathematical domain of limit theorems, which was proposed as a challenge to automated theorem proving by the late Woody Bledsoe. Using the proof planner of the OMEGA system we were able to solve all the well known challenge theorems including those that cannot be solved by any of the existing traditional systems.

Ilkka Niemelä, Helsinki University of Technology, Finland

Towards Declarative Rule-Based Constraint Programming

Normal logic programs allow an interesting constraint-oriented interpretation where rules of a program are seen as constraints on a solution set for the program. An intuitive declarative semantics for the solution sets is provided by the stable models of the program. This leads to a flexible and expressive rule-based constraint programming paradigm with logic programs and the stable model semantics. An efficient implementation method for the paradigm has

been developed and the key issues of the method are presented. An implementation of the method has been devised and the resulting system called Smodels is publicly available (see, http://www.tcs.hut.fi/pub/smodels/). Smodels is the fastest implementation of the stable model semantics currently available and the first system that is able to handle non-stratified ground programs with tens of thousands of rules. Encouraging experimental results on applying Smodels to combinatorial graph problems, planning and verification of distributed systems are discussed.

Robert Nieuwenhuis, Universitat Politecnica de Catalunya, Spain (joint work with M. Bofill, G. Godoy and A. Rubio)

Paramodulation modulo Weak Orderings

All current completeness results for ordered paramodulation require the term ordering ¿ to be well-founded, monotonic and total(izable) on ground terms. We introduce a new proof technique where the only properties required for ¿ are well-foundedness and the subterm property (this solves a well-known open problem, e.g., at the RTA list of open problems since 1995). The technique is a relatively simple and elegant application of some fundamental results on the termination and confluence of ground term rewrite systems (TRS).

By a careful further analysis of our technique, we obtain the first Knuth-Bendix completion procedure that finds a convergent TRS for a given set of equations E and a (possibly non-totalizable) reduction ordering ¿ whenever it exists (this was an even better known open problem, e.g., posed by N. Dershowitz and J-P. Jouannaud on the RTA list of open problems since its creation in 1991). Note that being a reduction ordering is the minimal possible requirement on ¿, since a TRS terminates if, and only if, it is contained in a reduction ordering.

Tobias Nipkow, Technische Universität München, Germany

Verified Lexical Analysis

This paper presents the development and verification of a (very simple) lexical analyzer generator that takes a regular expression and yields a functional lexical analyzer. The emphasis is on simplicity and executability. The work was carried out with the help of the theorem prover Isabelle/HOL.

Wolfgang Reif, Universität Ulm, Germany

Deductive Error Detection using KIV

Writing good formal specifications is a non-trivial task. In particular, formal modelling does not prevent developers from making errors. Therefore, in practical applications most of the time is not devoted to successful affirmative proofs (with respect to a formal model) but to failed proof attempts. Proof attempts fail because of errors in specifications, programs, lemmas, invariants etc. In the specification and verification system KIV [1], failed proof attempts are analysed automatically to detect the corresponding errors.

In KIV, error detection has two parts: finding a counter example to the current goal of the proof attempt, $\phi(\underline{x})$, and tracing it back to the earliest point of failure (by adapting the counter example). A counter example consists of a term assignment $\underline{x} = \underline{t}$ to the free variables (of generated sorts) of ϕ, and possibly some satisfiable residual formula, χ, restricting the class of counter models. χ and $\underline{x} = \underline{t}$ are deduced such that $\chi \wedge \underline{x} = \underline{t} \rightarrow \neg\, \phi(\underline{x})$ holds in all generated models of the specification additionally satisfying $\mathrm{Cl}_{\exists}\,(\chi)$. Backtracing the counter example towards the earliest point of failure delivers the proof step where the proof search most likely went wrong. If this is the root of the proof tree then the original conjecture is erroneous, an the counter example gives the witness.

References

[1] W. Reif, G. Schellhorn, K. Stenzel, and M. Balser. Structured specifications and interactive proofs with KIV. In W. Bibel and P. Schmitt, editors, *Automated Deduction—A Basis for Applications*. Kluwer Academic Publishers, 1998.

Manfred Schmidt-Schauß, Universität Frankfurt, Germany

Context Unification and Bounded Second Order Unification

Context unification and bounded second order unification are restrictions of second order unification where the number of holes in the instantiating terms is restricted. Contetx unification permits exactly one hole, n-bounded second order unification permits 0 to n holes. Bounded second order unification is decidable by an algorithm that uses a result on a bound on the exponent of periodicity, however, Makanin's decision procedure is not used. In a joint work with Klaus Schulz we showed that if at most two context variables

are permitted, and the number of first order variables is not restricted, then context unification is decidable. This maybe a step in tackling the open problem of decidability of general context unification.

Helmut Schwichtenberg, Universität München, Germany

Programming with proofs

It is well known that from a constructive proof one can extract a (functional) program. Several aspects of such an extraction procedure are discussed, with an emphasis on analyzing the complexity of the extracted program. In a joint paper with S. Bellantoni and K-H. Niggl, it is shown how to restrict recursion on notation in all finite types so as to characterize the polynomial time computable functions. The restrictions are obtained by enriching the type structure with the formation of types $!\sigma$, and by adding linear concepts to the lambda calculus.

John Slaney, Australian National University, Canberra, Australia

KRIPKE: 15 Years on

We revisit the work of the late Paul Thistlewaite on automated theorem proving for substructural logics. The theorem prover *KRIPKE*, due to Thistlewaite with R.K. Meyer and M.A. McRobbie, is based on a segment calculus presentation of the logic LR, which is the additive-multiplicative fragment of linear logic plus contraction. It incorporates several techniques for dealing with LR in the face of its extreme complexity (the decision problem is ESPACE-hard). We stress that logics of the kind addressed by *KRIPKE* are important for contemporary and future research, especially in AI, and note that some of the issues raised by Thistlewaite in the 1980s remain unresolved.

Viorica Sofronie-Stokkermans, Max-Planck-Institut für Informatik, Saarbrücken, Germany

Representation Theorems and Automated Theorem Proving in Varieties of Distributive Lattices with Operators

We present a method for automated theorem proving in the universal theory of certain varieties of distributive lattices with well-behaved operators. For this purpose, we use extensions of Priestley's representation theorem for distributive lattices.

We first establish a link between satisfiability of universal sentences with respect to varieties of distributive lattices with operators and satisfiability with respect to certain classes of relational structures. We then use these results for giving a method for translation to clause form of universal sentences in such varieties. The advantage is that we avoid the explicit use of the full algebraic structure of such lattices, instead using sets endowed with a reflexive and transitive relation and with additional functions and relations. Thus, the problems that occur when ACI-operators have to be considered (as is the case in algebraic automated reasoning for lattices) are avoided. Moreover, known saturation-based techniques for theories of reflexive and transitive relations, such as ordered chaining with selection, can be used successfully. Decidability and complexity results follow in many cases as consequences of existing decision procedures based on ordered resolution or ordered chaining. Finally, considerations concerning the structure of the sets of clauses generated with our method make certain algebraic properties of these varieties visible.

We first studied this type of relationships in the context of finitely-valued logics and then extended the ideas to more general non-classical logics. This paper shows that the idea is much more general. In particular, the method presented here subsumes both existing methods for translating modal logics to classical logic and methods for automated theorem proving in finitely-valued logics based on distributive lattices with operators.

Bruce Spencer and J.D. Horton, University of New Brunswick, Canada

Completeness, Uniqueness and Size Preserving Properties for Combinations of Restrictions of Resolution

We continue to study two restrictions of resolution from our presentation at the 1997 Dagstuhl seminar on Deduction, namely surgery-minimal and rank/activity. In this talk we investigate how the important properties of completeness, uniqueness and size preserving are affected when these restrictions are combined with some other restrictions from the literature: A-ordered, set of support, hyperresolution and subsumption. A restriction is said to exhibit uniqueness when it admits at most one element from each equivalence class imposed by the binary relation "resolves input literals similarly". Two binary resolution trees are said to resolve input literals similarly if they have the same multiset of input clauses and the same resolutions are done in each tree, although possibly in different orders. A restriction is said to be size preserving if it does not eliminate all smallest binary resolution trees, where size is number of nodes.

In addition we consider whether slight adjustments can make combinations of restrictions compatible. Many results are presented, including that the combination of set of support (SOS), rank/activity and surgery-minimal preserves completeness and uniqueness and preserves the size of the smallest SOS binary resolution tree, provided that the following adjustment is made: in the resolvent of a SOS clause against a non-SOS clause, the literals from the non-SOS clause are all active. A similar adjustment ensures the preservation of completeness and uniqueness by the the combination of hyperresolution, rank/activity and surgery-minimal, and furthermore the smallest hyperresolution tree is preserved. We conjecture that completeness and uniqueness is preserved when a similar adjustment is made to the combination of A-ordered resolution with hyperresolution and to A-ordered with set-of-support.

Michael Thielscher, TU Dresden, Germany

Applied Deduction: Cognitive Robotics

Being a central aspect of Artificial Intelligence, research in Cognitive Robotics aims at endowing robots with high-level cognitive functions. These enable robots to reason about tasks and goals, about perceptions, and about actions that can be taken and their effects on the environment. Robots with this capability do not function by executing a rigid and detailed algorithm. Rather they devise plans on their own on the basis of information about the goal and the current situation. Thus they are able to adapt to different tasks and environments and therefore allow very flexible usage. The classical approach towards the general goal of Cognitive Robotics assumes intelligent behavior in a dynamic world to be a result of correct reasoning on correct representations. In turn, this reasoning is understood by means of formal logic.

Ashish Tiwari, State University of New York, Stony Brook, USA (Joint work with Leo Bachmair)

Abstract Congruence Closure and Applications

Congruence closure algorithms for solving word problems for finitely presented algebras have also been used in combining decision procedures. On the other hand, congruence closure can itself be looked upon as a combination problem. Taking this view leads us to define the notion of an *abstract congruence closure*. We present a completion based description of how such closures can be constructed. Congruence closure algorithms that appear in

the literature (Downey-Sethi-Tarjan, Nelson-Oppen and Shostak) can be seen as specific implementations of this abstract completion.

This abstract view of congruence closure has added advantages. Apart from giving a better understanding, it helps us to come up with efficient algorithms for congruence closure. It is easily extended to include *AC* function symbols. This gives us completion based transition rules for computing an *AC* congruence closure. Congruence closure has also been used in construction of ground convergent systems, and in performing efficient normalization. These applications are also simplified and generalized in this framework. Of particular interest is the problem of construction of ground AC convergent system using AC congruence closure.

Tomás E. Uribe, Stanford University, USA

Abstraction, Validity Checking, and Model Checking

Proving temporal properties of general infinite-state reactive systems is undecidable. However, *deductive verification* reduces temporal properties of a system to the general validity of first-order verification conditions, which can be established using theorem proving. This results in verification methods that are complete, relative to the expressiveness of the assertion language used [MP95]. For particular classes of reactive systems, temporal properties can be algorithmically *model checked* if the model (Kripke structure) of the system can be finitely described and systematically explored. This is most often the case with relatively small finite-state systems.

Many deductive verification methods can be viewed as model checking a small finite-state *abstraction* of the system, whose correctness is deductively justified. Furthermore, validity checking can be used not only to prove the correctness of given abstractions, but to *generate* abstractions as well, e.g. [GS97, CU98]. In this *deductive-algorithmic verification*, validity checking is used to reason about the possibly unbounded domains of the concrete system. For this, specialized decision procedures must continue to be developed, combined, and integrated into first-order provers, e.g. [BSU97].

Joint work with Zohar Manna and the STeP (Stanford Temporal Prover) group: Nikolaj Bjørner, Anca Browne, Michael Colón, Bernd Finkbeiner and Henny Sipma.

References

[BSU97] Nikolaj S. Bjørner, Mark E. Stickel, and Tomás E. Uribe. A practical integration of first-order reasoning and decision procedures. In

Proc. of the 14^{th} *Intl. Conference on Automated Deduction*, volume 1249 of *LNCS*, pages 101–115. Springer-Verlag, July 1997.

[CU98] Michael A. Colón and Tomás E. Uribe. Generating finite-state abstractions of reactive systems using decision procedures. In Alan J. Hu and Moshe Y. Vardi, editors, *Proc.* 10^{th} *Intl. Conference on Computer Aided Verification*, volume 1427 of *LNCS*, pages 293–304. Springer-Verlag, July 1998.

[GS97] Susanne Graf and Hassen Saidi. Construction of abstract state graphs with PVS. In Orna Grumberg, editor, *Proc.* 9^{th} *Intl. Conference on Computer Aided Verification*, volume 1254 of *LNCS*, pages 72–83. Springer-Verlag, June 1997.

[MP95] Zohar Manna and Amir Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer-Verlag, New York, 1995.

Robert Veroff, University of New Mexico, USA

Reasoning at Multiple Levels of Abstraction

In order to be successful in complex domains involving hierarchies of defined terms, an automated reasoning program must be able to reason effectively at multiple levels of abstraction. At a minimum, this requires appropriate problem representations and good search strategies. Ideally, the reasoning program reasons at high levels of abstraction when possible and appeals to arguments at lower levels of abstraction only when necessary.

In this talk, we describe our early experiences developing representations and search strategies for an application of automated reasoning to a problem domain from theoretical computer science. We then summarize some of the approaches we are developing to permit the automated reasoning program to reason effectively at multiple levels of abstraction. These approaches include a search strategy, the use of a special class of inference rules, and distributed theorem proving.

Andrei Voronkov, University of Manchester, Great Britain

Deciding propositional K by the inverse method.

Nonclassical (propositional) logics play an increasing role in computer science. They are used in model checking, verification, and knowledge representation. Traditional decision procedures for these logics are based on semantic tableaux, SAT-based methods, or translation into classical predicate logic.

We present a bottom-up decision procedure for propositional modal logic K based on the inverse method. In order to make the procedure efficient we prove a number of redundancy criteria for derivations in a sequent calculus. A new presentation of the inverse method using a *path calculus* is given and a new technique for proving redundancy criteria is presented, based on analysis of tableau-based derivations in K. We also present experimental results showing that our decision procedure can complement tableau-based procedures.

Christoph Weidenbach, MPI für Informatik, Saarbrücken, Germany

Towards an Automatic Analysis of Security Protocols in First-Order Logic

The growing importance of the internet causes a growing need for security protocols that protect transactions and communication. It turns out that the design of such protocols is highly error-prone. Therefore, a variety of different methods have been described that analyze security protocols to discover flaws. In my talk I describe a new method that is based on automated theorem proving in first-order logic. Using some example protocols, I show how security aspects can be formalized in first-order logic, such that the resulting formulae are subject to automated theorem proving. In addition to the analysis, I develop the necessary theoretical background providing new (un)decidability results for the first-order fragments involved in the analysis and I identify possible extensions leading to future directions of research.

Victor L. Winter, Sandia National Laboratories, Albuquerque, USA

Grammar Oriented Program Transformation

Program transformation is a technique that can be used to derive a software implementation from a formal specification. From the perspective of correctness, the transformational process should proceed in steps that are small enough that they can be formally (and automatically) verified. Unfortunately, this "size" constraint places severe limitations on the ability to apply transformations manually. Thus a need for automatic control arises. However, in order to be practical, an automatic control paradigm must be rich enough to permit intelligent control strategies to be realized in a straightforward manner.

In this talk, I will give an overview of some of the control mechanisms that are supported by the High Assurance Transformation System (HATS) developed at Sandia National Laboratories.